

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: Detection In The Presence Of
Media Noise

APPLICANT: GREGORY BURD

CERTIFICATE OF MAILING BY EXPRESS MAIL

EU328702231US

Express Mail Label No.

October 31, 2003

Date of Deposit

DETECTION IN THE PRESENCE OF MEDIA NOISE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part of U.S. non-provisional application No. 10/208,312, filed July 29, 2002, which claims the benefit of the filing date of U.S. provisional application No. 60/345725 filed January 3, 2002, the contents of which are herein incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] An aspect of this invention relates to decoding a communication signal in the presence of noise.

BACKGROUND

[0003] With the continuing evolution of computer systems, there is an increasing demand for greater storage density. But, as storage density increases, problems associated with signal dependent noise and interference increase. To detect data in the presence of signal dependent noise, detectors typically employ complex schemes such as modifying the Euclidean branch metric to compensate for the noise and adaptively computing channel statistics. Other detectors have used a post-processor based on a model of the channel.

SUMMARY

[0004] A signal detector to detect data in an input signal. The signal detector includes a finite impulse response (FIR) filter to equalize the data to a primary target. A Viterbi-like detector is matched to the primary target and generates a most likely path corresponding to the data in the input signal. A linear post-processor determines at least one most likely error event in the most likely path and generates revised paths based on the at least one most likely error event. A media noise processor operates on the data with a secondary target that is different from the primary target. The media noise processor computes path metrics corresponding to each of the revised paths as a function of a non-linear noise model and selects one of the revised paths based on the path metrics.

[0005] The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0006] FIG. 1 illustrates a recording assembly for storing data.

[0007] FIG. 2 shows a receiver for receiving a communication signal.

[0008] FIG. 3 is a block diagram of an aspect of a communication signal detector.

[0009] FIG. 4 is a block diagram of an aspect of a communication signal detector.

[0010] FIG. 5 is a block diagram of an aspect of an error event filter.

[0011] FIG. 6 is an error event processing diagram of a block of code.

[0012] FIG. 7 is a flow diagram of a process for detecting data in a communication signal.

[0013] FIG. 8 is a block diagram of an aspect of a communication signal detector.

[0014] FIG. 9 is a block diagram of an aspect of a communication signal detector.

[0015] FIG. 10 is a block diagram of an aspect of a communication signal detector.

[0016] FIG. 11 is a graphical representation of primary and secondary targets.

[0017] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0018] Figure 1 shows a recording assembly 10 for storing data. The recording assembly 10 includes media 12 to which data may be written and read. The media 12 may be included in any known storage device such as hard drives and optical disks. In a writing operation, a signal processor 14 may apply compression and error correction schemes to an input signal. An encoder 16 converts the processed input signal to a format suitable for storage by the storage unit 12. A preamp 18 amplifies and writes the encoded signal to the media 12. In a reading operation, a read head amplifier 20 detects and generates the read back signal by reading data from the media 12. The read head amplifier 20 may include a read equalizer for equalizing the data. A detector 22 detects symbols in the read back signal. The detector 22 is particularly suitable for signal detection in the presence of signal dependent noise. Although the detector 22 is described in conjunction with a recording system, the detector may be used for signal detection over any type of communication channel. A decoder 21 may decompress the decoded signal. The signal processor 14 may apply error correction to the decompressed signal to generate an output signal representing the recovered data.

[0019] Figure 2 shows a receiver 30 for decoding a communication signal that may include signal dependent noise. A

signal processor 32 filters and equalizes the communication signal. The data in the filtered communication signal may be detected by a signal detector 34 in accordance with the teachings of the invention. The signal detector 34 is not limited to being implemented in the illustrated receiver 30, instead the signal detector may be included in any appropriate receiver. An RLL/ECC decoder 36 may then decode the detected data.

[0020] Figure 3 shows an aspect of a non-linear signal detector 40 for detecting data in a communication signal. The non-linear signal detector 40 may detect data in the presence of media noise such as inter-symbol interference and other pattern dependent noise. A channel 42 communicates a stream of data to the non-linear signal detector 40. A signal processor 44 may filter and equalize the stream of data. A finite impulse response (FIR) filter 46 filters the stream of data. A Viterbi-like detector 48 may detect data in the stream of data and generate soft or hard decisions. Any Viterbi-like detector 48 that is derived under the assumption of additive white Gaussian noise (AWGN) may be employed such as partial response maximum likelihood (PRML) schemes or hybrids between tree/trellis detectors and decision feedback equalizers (DFE) including Fixed Delay Tree Search with Decision Feedback (FDTS/DF), and multilevel decision feedback (MDFE). A linear post-processor 50

filters the Viterbi decisions for dominant error events by identifying the least reliable Viterbi decisions. A list of the typical dominant error events may be used to identify the least reliable Viterbi decisions. The typical dominant error event list may include dominant error events such as {+}, {+-}, {++}, {+--}, and {+---}. For each block of length C, the linear post-processor 50 may identify a list of N least reliable Viterbi decisions. The list of least reliable decisions may include decision information such as error location, error type, and polarity. The linear post-processor 50 may evaluate a metric such as a linear maximum likelihood distance penalty (MLDP) associated with an error event to identify the least reliable Viterbi decisions. The most likely error events may be determined based on the list of typical dominant error events and a computed path metric. Preferably, two most likely error events are determined. However, any number of error events may be selected ranging from at least one. In addition, the number of error events selected may be varied based on factors such as block length C and the frequency of low reliability decisions. Revised paths corresponding to each of the least reliable decisions may be generated based on the most likely error events.

[0021] The non-linear MLDP may be derived from a non-linear noise model of the communications channel output. A non-linear

post-processor 52 may determine a non-linear MLDP for each of the revised paths. The lower the value of the non-linear MLDP ($MLDP_{nlin}$), the more likely the corresponding path is the best path. The non-linear MLDP may be a function of a non-linear path metric that includes a representation of data dependent channel noise.

$$MLDP_{nlin}(error) = PM_{nlin}(viterbi_out + error) - PM_{nlin}(viterbi_out) =$$

$$= \sum_{Branches} (BM_{nlin}(viterbi_out + error) - BM_{nlin}(viterbi_out))$$

[0022] A noise estimation block 54 may estimate the noise characteristics by using a non-linear channel model such as that described below in Equation BM, where $x(D)$ represents the input bits to a channel and $y(D)$ represents the output values of the channel.

$$y_t = \bar{y} + m_t(x(D)) + n_t(x(D))$$

$$\text{where; } \bar{y}_t = \sum_{k=0}^L x_{t-k} \cdot h_k, \quad m_t(x(D)) \text{ represents data dependent mean}$$

shift, $n_t(x(D))$ is modeled as signal dependent Gauss-Markov noise process with memory length L , and h_k represents a primary equalization target.

$$n_t(x(D)) = \sum_{k=1}^L f_k(x(D)) n_{t-k} + \sigma_t(x(D)) N(0,1) \quad , \text{ To simplify,}$$

it is further assumed that;

$$m_i(x(D)) = m_i(x_{i-1}, \dots, x_{i-2}, x_{i-1}, x_i)$$

$$\sigma_i(x(D)) = \sigma_i(x_{i-1}, \dots, x_{i-2}, x_{i-1}, x_i)$$

$$f_k(x(D)) = f_k(x_{i-1}, \dots, x_{i-2}, x_{i-1}, x_i), \text{ for } k = 1, \dots, L$$

The corresponding non-linear Viterbi Branch Metric (BM) is given by:

$$BM = \ln(\sigma_i^2(x(D))) + \left(y_i - (\bar{y}_i + m_i) - \sum_{k=1}^L f_k(x(D)) n_{i-k} \right)^2 \left(\frac{1}{\sigma_i^2(x(D))} \right)$$

$$BM = \ln(\sigma_i^2(x(D))) + \left(n_i - \sum_{k=1}^L f_k(x(D)) n_{i-k} \right)^2 \left(\frac{1}{\sigma_i^2(x(D))} \right)$$

$$BM = \ln(\sigma_i^2(x(D))) + \left(\sum_{k=0}^L \hat{f}_k(x(D)) n_{i-k} \right)^2 \quad \text{Eq. BM}$$

$$\text{where; } \hat{f}_0 = \frac{1}{\sigma(x(D))}, \hat{f}_1 = -f_1 \cdot \frac{1}{\sigma(x(D))}, \dots, \hat{f}_3 = -f_3 \cdot \frac{1}{\sigma(x(D))}$$

[0023] A selector 56 selects the revised path that represents the most likely path based upon the path metrics such as the non-linear MLDP computed for each of the most likely error events associated with a least reliable Viterbi decision. The most likely path is the revised path having the lowest non-linear path metric. The selection of the most likely path may

also be a function of other path information such as parity error status.

[0024] A correction block 58 revises the Viterbi decision to correspond to the selected most likely path and inserts the revised Viterbi decision into the output data stream.

[0025] Figure 4 shows another aspect of a non-linear signal detector 60 for detecting data in a communication signal. The non-linear signal detector 60 is similar in function to the non-linear signal detector 40 and including a Viterbi detector 62 to detect symbols in a data stream and generate decisions. A linear post-processor 64 and a non-linear post-processor 66 operate similarly to the linear post-processor 50 and non-linear post-processor 52. A channel reconstruction filter 68 may filter the Viterbi decisions to generate noiseless samples. The outputs of the delay 70 and channel reconstruction filter 68 may be subtracted to determine linear noise/error samples which are input to the linear post-processor 64. An error feasibility check block 72 may evaluate the Viterbi decisions and linear post-processor outputs to determine whether an error might exist in the Viterbi decisions for a codeword. A parity block 74 determines the parity status, violation or no violation, of codewords of the Viterbi decisions. A correction block 76 may evaluate the parity status, then determine whether to select one of the revised paths or pass the Viterbi decisions for the

codeword through without corrections. For example, the threshold for the MLDP value at which a codeword will be corrected may be adjusted as a function of the parity status so that when a parity violation is indicated, the Viterbi decisions may be corrected even though the computed MLDPs do not strongly indicate a most likely path. Alternately, when a parity violation is not indicated, the Viterbi decisions for the codeword will not be corrected unless the computed MLDPs very strongly indicate a most likely path.

[0026] Figure 5 shows a block diagram of an aspect of a non-linear error event filter 80. The non-linear error event filter 80 may compute an $MLDP_{nlin}$ based on the non-linear path metric described in Equation BM. The $MLDP_{nlin}$ may be described by the following equation.

$$MLDP_{nlin}(error) = PM_{nlin}(viterbi_out + error) - PM_{nlin}(viterbi_out) \quad \text{Eq. 1}$$

[0027] Figure 6 shows an error event processing diagram 90 of a block of code that may be processed by the non-linear error event filter 80. The error event processing diagram 90 illustrates the interrelationship between a Viterbi output 92, a 5-bit error 94, a resulting Viterbi+error 96, and a noise sample 98 corresponding to the Viterbi output 92.

[0028] Figure 7 shows a flow diagram of a process for detecting data in a communication signal. Starting at block 100, data in an input signal is detected by a Viterbi-like

detector. Decisions, hard or soft, based on the detected data are generated at block 102. At block 106, the most likely error events are determined. Continuing to block 108, revised paths are generated based on the most likely error events. At block 110, path metrics corresponding to each of the revised paths are computed. The path metrics may be used to compute an MLDP corresponding to each of the revised paths. At block 112, one of the revised paths is selected based on the path metrics.

[0029] Figure 8 shows a nonlinear signal detector 200 for detecting data in a communication signal. The nonlinear signal detector 200 may detect data in the presence of noise such as intersymbol interference (ISI) and other pattern dependent noise. An analog-to-digital converter (ADC) 202 may process an analog stream of data into a digital stream of data. A FIR filter 204 filters the stream of data. The FIR filter may equalize the data to a primary target. In one aspect the primary target may be a 5 tap target. A Viterbi-like detector 206 may detect data in the stream of data and generate hard or soft decisions. The Viterbi-like detector 206 may be matched to the primary target. The complexity of the Viterbi-like detector may be decreased by limiting the target length. In one example, where the FIR filter 204 may equalize to a 5 tap target, the Viterbi-like detector 206 may be matched to the 5 tap target length. Any type of Viterbi-like detector that is derived under

the assumption of AWGN may be employed such as PRML schemes, hybrids between tree/trellis detectors, DFES including FDTS/DF, and multilevel decision feedback.

[0030] A linear post-processor 208 may filter the Viterbi decisions for likely error events by identifying the least reliable Viterbi decisions. The linear post-processor 208 may function in a manner similar to the linear post-processor 50 described above and shown in Figure 3. The linear post-processor 208 may be matched to either the primary target or a secondary target that may be a different target than that of the Viterbi detector 206.

[0031] A media noise processor 210 may compute path metrics for one or more of the revised paths determined by the linear post-processor 208. Then, the media noise processor may select one or more of the revised paths based on the computed path metrics. The path metrics may be computed as a function of a non-linear noise model that may model noise associated with the media on which the data is stored and communicated over. Any type of non-linear noise model may be employed. In one aspect, the non-linear noise model may include a channel response estimator 209 to estimate the channel response and a noise characteristics estimator 211 to estimate the noise characteristics. In another aspect, the media noise processor 210 may determine a non-linear MLDP as in Eq. 1. The lower the

value of the non-linear MLDP, the more likely the corresponding path is the best path. The non-linear MLDP may be a function of a non-linear path metric that includes a representation of data dependent channel noise. The media noise processor 210 advantageously may use a different target (secondary target) than the target to which the Viterbi-like detector 206 is matched (primary target). The secondary target and primary target may differ in any parameter such as target length, tap values, and the number of taps. In one aspect, the secondary target of the media noise processor 210 may be longer than the primary target, which may lead to an increase in the accuracy of the path metrics that the media noise processor 210 computes. Although an increased target length may lead to an increase in the complexity of the media noise processor 210, the complexity only increases at a linear rate with increasing target length. For the Viterbi on the other hand, the complexity grows exponentially with increasing target length. By using a secondary target for the media noise processor 210 the accuracy of the computed path metrics may be increased, with only a minimal increase in the complexity of the media noise processor 210 and the linear post-processor 208, and no increase in the complexity of the Viterbi.

[0032] The secondary target used by the media noise processor 210 and possibly the linear post processor 208 is not known

apriori and hence needs to be estimated. Until the secondary target estimates are available, the linear post-processor 208 and media noise processor 210 may operate with the primary target.

[0033] A noise estimation block 212 may receive the data from the FIR filter 204 and the Viterbi-like detector 206, and estimate an equalized channel response and media noise characteristics. For example, the channel noise may be estimated based on a non-linear channel model such as that described above in equation BM, although any non-linear channel model may be employed. The noise estimation block 212 may also estimate non-linear parameters such as mean shifts, noise whitening filtering (NWF), and residual noise variance. The noise estimation block 212 may track the secondary target. The secondary target response is initialized and aligned with the primary target of the Viterbi-like detector 206. In one aspect, a least mean square engine may be used to adapt the secondary target response, although any type of adaptive procedure may be used to adapt the secondary target response.

[0034] Figure 11 shows a graphical representation of an exemplary primary target 220 and a secondary target 222 adapted from the output of the FIR filter 204. The secondary target 222 is similar to the primary target 220 and is not limited to a response that has the same or fewer taps as the primary target

220, such as zero outside of the five taps in this example.

Also, the taps of the secondary target are not limited to being integer valued, and therefore may have improved resolution and less mis-equalization.

[0035] A linear error signal generator 214 may compute the linear error. The linear error signal generator 214 may include a channel reconstruction filter 213 to convolve the output of the Viterbi-like detector 206 with the secondary target taps from the noise estimation block 212. A delay 217 may delay the output from the FIR filter 204. The linear error signal may subtract the the output of the channel reconstruction filter 213 from the output of the FIR filter 204. The linear error signal generator 214 may communicate the linear error to the linear post-processor 208 and a nonlinear filter 216. The linear error generated by the linear error signal generator 214 is based on the secondary target.

[0036] In another aspect, another linear error signal generator 215 may compute a linear error by subtracting the output of the Viterbi-like detector 206 from the output of the FIR filter 204. The linear error signal generator 215 may include a channel reconstruction filter, delay, and subtracter configured similarly to the linear error signal generator 214. The linear error signal generator 215 may communicate the linear error to the linear post-processor 208. The linear error

computed by the linear error signal generator 215 is based on the primary target. In this aspect, the computed linear error of the linear error signal generator 214 is preferably only communicated to the nonlinear filter 216.

[0037] The nonlinear filter 216 may compute the non-linear Viterbi Branch Metrics corresponding to each of the revised paths that are determined by the linear post-processor 208.

[0038] A correction block 218 may select the revised path that represents the most likely path based upon the path metrics computed by the non-linear filter for each of the most likely error events associated with a least reliable Viterbi decision. The most likely path may be the revised path having the lowest non-linear path metric. The selection of the most likely path may also be a function of other path information such as parity error status. The correction block 218 may revise the decision of the Viterbi-like detector 206 to correspond to the selected most likely path and insert the revised Viterbi decision into the output data stream.

[0039] Figure 9 shows another nonlinear signal detector 300 similar in function to nonlinear signal detector 200 and with corresponding elements in the range 300-320, except that in nonlinear signal detector 300 the media noise processor 310 operates on data that has not been filtered by the FIR filter 304. By operating on data prior to filtering by the FIR filter

304, the nonlinear signal detector 300 minimizes media noise characteristics degradation caused by the FIR filter 304 that may degrade the computation of the branch metrics. The estimated channel response following the ADC 202 is likely to be longer than the response following the FIR filter 204, since the response following the FIR filter 204 is usually limited in size to reduce the complexity of the Viterbi-like detector 206. The delay 317 may be set based on the channel implementation, for example the delay 317 may be longer than the delay 217.

[0040] Figure 10 shows a linear signal detector 400 for detecting data in a communication signal. An analog-to-digital converter (ADC) 402 may process an analog stream of data into a digital stream of data. A FIR filter 404 may filter the stream of data. The FIR filter 404 may equalize the data to a primary target such as a 5 tap target length. A Viterbi-like detector 406 may detect data in the stream of data and generate hard or soft decisions. The Viterbi-like detector 406 may be matched to the primary target. The complexity of the Viterbi-like detector 406 may be decreased by limiting the primary target length. In one example, where the FIR filter 404 may equalize to a 5 tap target and the Viterbi-like detector 406 may be matched to the 5 tap target. Any type of Viterbi-like detector that is derived under the assumption of AWGN may be employed such as PRML

schemes, hybrids between tree/trellis detectors, DFEs including FDTs/DF, and multilevel decision feedback.

[0041] In one aspect, a channel response estimator 408 may advantageously track a secondary target that is different than the primary target. The secondary target may differ from the primary target in any parameter such as target length, the value of the target taps, and the quantity of target taps. In one aspect, the secondary target may be selected to be longer than the primary target. The data to the channel response estimator 408 may be communicated from the output of the FIR filter 404. The secondary target response is initialized and aligned with the target response of the Viterbi-like detector 406. The channel response estimator 408 may include a least mean square engine 410 to adapt the secondary target. In another aspect, the data may also be communicated from the ADC 402. Also, a multiplexer 409 may be employed to select between the output of the FIR filter 404 and the output of the ADC 402.

[0042] A linear post-processor 412 may filter the Viterbi decisions for dominant error events by identifying the least reliable Viterbi decisions. The linear post-processor 412 may advantageously be matched to either the secondary target or the primary target. For example, if a secondary target is used that is longer than the primary target, mis-equalization may be reduced. Although the increased target length of the secondary

target may lead to an increase in the complexity of the linear post-processor 412, the complexity only increases at a linear rate with increasing target length. For the Viterbi-like detector 406 on the other hand, the complexity grows exponentially with increasing target length. By using a longer target length for the linear post-processor 412 than the target length used for the Viterbi detector 406, the accuracy of the computed path metrics may be increased, with only a minimal increase in the complexity of the linear post-processor 412 and no increase in the complexity of the Viterbi 406. In one aspect, either only the output from the ADC 402 or only the output from the FIR filter 404 may be communicated to the linear post-processor 412. In another aspect, a multiplexer 413 may be used to select one of the output from the ADC 402 or the output from the FIR filter 404. The multiplexer 413 and the multiplexer 409 preferably select each select the same output from either the FIR filter 404 or the ADC 402.

[0043] A linear error signal generator 414 may compute a linear error. The linear error signal generator 414 may include a channel reconstruction filter to shape the output of the Viterbi-like detector 406. Either the output of the ADC 402 or the output of the FIR filter 404 may be delayed and then subtracted from the output of the channel reconstruction filter to generate the linear error. The source of the data to the

linear error signal generator 414 is preferably the same as the source of data for the channel response estimator 408.

[0044] A linear filter 416 may filter the Viterbi decisions for dominant error events by identifying the least reliable Viterbi decisions. A list of the typical dominant error events may be used to identify the least reliable Viterbi decisions. For each block of length C , the linear filter 416 may identify a list of N least reliable Viterbi decisions. The list of least reliable decisions may include decision information such as error location, error type, and polarity. The linear filter 416 may compute path metrics from the linear error for the corresponding Viterbi decisions to identify the least reliable Viterbi decisions. The most likely error events corresponding to each of the least reliable decisions may be determined based on the list of typical dominant error events and the computed path metric. Preferably, two most likely error events are determined. However, any number of error events may be selected ranging from at least one. In addition, the number of error events selected may be varied based on factors such as the frequency of low reliability decisions. Revised paths corresponding to each of the least reliable decisions may be generated based on the most likely error events

[0045] A correction block 418 may select the revised path that represents the most likely path based upon the computed

path metrics for each of the most likely error events corresponding to a least reliable Viterbi decision. The most likely path is the revised path having the lowest non-linear path metric. The selection of the most likely path may also be a function of other path information such as parity violation status. A parity check block 420 may determine the parity violation status. The correction block 218 may revise the decision of the Viterbi-like detector 206 to correspond to the selected most likely path and insert the revised Viterbi decision into the output data stream.

[0046] A number of aspects of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.